

---

Masters Theses

Student Theses and Dissertations

---

1966

## A comparison of the LR and QR transformations for finding the eigenvalues for real nonsymmetric matrices

Susan Clara Hanson

Follow this and additional works at: [https://scholarsmine.mst.edu/masters\\_theses](https://scholarsmine.mst.edu/masters_theses)



Part of the [Computer Sciences Commons](#)

Department:

---

### Recommended Citation

Hanson, Susan Clara, "A comparison of the LR and QR transformations for finding the eigenvalues for real nonsymmetric matrices" (1966). *Masters Theses*. 5740.

[https://scholarsmine.mst.edu/masters\\_theses/5740](https://scholarsmine.mst.edu/masters_theses/5740)

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact [scholarsmine@mst.edu](mailto:scholarsmine@mst.edu).

A COMPARISON OF THE  
LR AND QR TRANSFORMATIONS FOR  
FINDING THE EIGENVALUES  
OF REAL NONSYMMETRIC MATRICES

BY

SUSAN CLARA HANSON

---

A

THESIS

submitted to the faculty of the

UNIVERSITY OF MISSOURI AT ROLLA

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Rolla, Missouri

1966

---

Approved by

Ralph E. Lee (Advisor)

Billy E. Gillett

John D. M. Wellish

Herbert R. Alcorn

## ABSTRACT

The LR and QR algorithms, two of the best available iterative methods for finding the eigenvalues of a nonsymmetric matrix associated with a system of linear homogeneous equations, are studied. These algorithms are discussed as they apply to the determination of the eigenvalues of real nonsymmetric matrices.

A comparison of the speed and accuracy of these transformations is made. A detailed discussion of the criterion for convergence and the numerical difficulties which may occur in the computation of multiple and complex conjugate eigenvalues are included.

The results of this study indicate that the QR algorithm is the more successful method for finding the eigenvalues of a real nonsymmetric matrix.

## ACKNOWLEDGEMENT

The author wishes to express her sincere appreciation to Professor Ralph E. Lee, Director of the Computer Science Center, for his help in the selection of this subject and for his guidance and suggestions during the preparation of this thesis.

## TABLE OF CONTENTS

	Page
ABSTRACT . . . . .	ii
ACKNOWLEDGEMENT . . . . .	iii
I. INTRODUCTION . . . . .	1
II. REVIEW OF LITERATURE . . . . .	3
III. DISCUSSION AND COMPARISON OF ALGORITHMS . . . . .	24
IV. RESULTS AND CONCLUSIONS . . . . .	32
APPENDIX A. Programs . . . . .	35
APPENDIX B. Numerical Results . . . . .	42
BIBLIOGRAPHY . . . . .	47
VITA . . . . .	48

## I. INTRODUCTION

Iterative methods for finding all the eigenvalues of nonsymmetric matrices have appeared only recently. The LR algorithm, which can be regarded as a development of the QD algorithm, was introduced by Rutishauser (9). It is based on the successive triangular decomposition of a matrix. A sequence of similar matrices is generated whose limit is triangular. A modified procedure for the LR algorithm improves numerical stability. It uses a modified decomposition with interchanges.

The QR algorithm, later developed by Francis (2), (3), makes use of unitary transformations instead of triangular decomposition. A variation of this algorithm involves a double-shift technique for combining complex conjugate shifts of origin while using Householder's method. It was developed for finding complex conjugate eigenvalues of real matrices and is known as the double QR procedure.

The LR and QR transformations are applicable to the calculation of general matrices, but a large number of computational operations are required. Therefore, it is desirable to use a matrix of condensed form. Both of these transformations preserve the form of a Hessenberg or almost triangular matrix.

The preliminary reduction of a matrix to Hessenberg form can be accomplished in several ways. However, in this paper we will be concerned with two methods: The Householder transformation using elementary orthogonal matrices and Gaussian elimination using stabilized elementary matrices.

The purpose of this study is to compare the LR and QR transformations for finding eigenvalues of real nonsymmetric matrices. Categories of

matrices which may cause the algorithm to fail are discussed. Shifts of origin and deflation are incorporated to improve convergence. Computational aspects, such as speed and accuracy are discussed, and a number of sample computations using these algorithms are cited.

## II. REVIEW OF LITERATURE

The LR algorithm for finding all the eigenvalues of an arbitrary matrix, introduced by Rutishauser (9) in 1955, is an interpretation of his QD scheme. Its basis is the triangular decomposition of a matrix. The matrix  $A_k$  is factorized into the product of a unit left triangular matrix  $L$  and a right triangular matrix  $R$  such that

$$LR = A \quad (2.1)$$

The LR method begins with the original matrix  $A_1$ . A sequence of matrices  $A_k$  is then formed such that  $A_{k+1}$  is derived from  $A_k$  by decomposing it into  $L_k$  and  $R_k$  and forming the product of these in reverse order.

$$A_k = L_k R_k \quad (2.2)$$

$$A_{k+1} = R_k L_k \quad \text{for } k=1, 2, \dots$$

In this process, a series of similarity transformations are performed on the original matrix  $A_1$  each of which consists of premultiplication by a matrix which eliminates the subdiagonal elements and post-multiplication by its inverse.

$$\begin{aligned} A_k &= R_{k-1} L_{k-1}^{-1} A_{k-1} L_{k-1} \\ &= L_{k-1}^{-1} \dots L_2^{-1} L_1^{-1} A_1 L_1 L_2 \dots L_{k-1} \end{aligned} \quad (2.3)$$

Equation (2.3) can be rewritten as

$$L_1 L_2 \dots L_{k-1} A_k = A_1 L_1 L_2 \dots L_{k-1} \quad (2.4)$$

Using  $A_k = L_k R_k$  and (2.4) it follows that

$$\begin{aligned} L_1 L_2 \dots L_k R_k &= A_1 L_1 L_2 \dots L_{k-1} \\ L_1 L_2 \dots L_{k-1} R_{k-1} &= A_1 L_1 L_2 \dots L_{k-2} \end{aligned}$$

and so on.



If

$$T_k = L_1 L_2 \dots L_k \quad (2.5)$$

and

$$U_k = R_k \dots R_2 R_1$$

where  $T_k$  is unit left triangular and  $U_k$  is right triangular, then

$$\begin{aligned} T_k U_k &= L_1 L_2 \dots L_{k-1} (L_k R_k) R_{k-1} \dots R_2 R_1 \\ &= L_1 L_2 \dots L_{k-1} A_k R_{k-1} \dots R_2 R_1 \\ &= A_1 L_1 L_2 \dots L_{k-1} R_{k-1} \dots R_2 R_1 \\ &= A_1^2 L_1 L_2 \dots L_{k-2} R_{k-2} \dots R_2 R_1 \\ &= A_1^k. \end{aligned} \quad (2.6)$$

The triangular decomposition of  $A_1^k$  is  $T_k U_k$ . All of the matrices  $A_k$  have the same eigenvalues since they are similar. In a proof of convergence Rutishauser (9) shows that, under certain conditions,  $A_k$  tends to a right triangular matrix as  $k \rightarrow \infty$ . The diagonal elements of this right triangular matrix are the eigenvalues of  $A_k$  appearing in decreasing order of magnitude from left to right.

The process of triangular decomposition is discussed by Faddeev and Faddeeva (1), Ralston (8), and Wilkinson (13). The problem is that of determining the matrices  $L$  and  $R$  of (2.1) directly from  $A$  without going through any intermediate steps. Wilkinson (13) assumes that the first,  $(r-1)$  rows of  $L$  and  $R$  can be determined by equating the elements in the first  $(r-1)$  rows of both sides of equation (2.1). The elements in the  $r^{\text{th}}$  row are equated so that

$$\left. \begin{aligned} d_{r1} u_{11} &= a_{r1} \\ d_{r1} u_{12} + d_{r2} u_{22} &= a_{r2} \\ d_{r1} u_{1r} + d_{r2} u_{2r} + \dots + d_{rr} u_{rr} &= a_{rr} \end{aligned} \right\} \quad (2.7)$$

$$\left. \begin{aligned} d_{r1} u_{1,r+1} + d_{r2} u_{2,r+1} + \dots + d_{rr} u_{r,r+1} &= a_{r,r+1} \\ d_{r1} u_{1n} + d_{r2} u_{2n} + \dots + d_{rr} u_{rn} &= a_{rn} \end{aligned} \right\} \quad (2.8)$$

The elements  $d_{r1}, d_{r2}, \dots, d_{r,r-1}$  are uniquely determined from equations 1 to  $(r-1)$  of (2.7). Since  $L$  is unit left triangular, then  $d_{rr} = 1$ , and the  $r^{\text{th}}$  equation determines  $u_{rr}$ . Then  $u_{r,r+1}$  to  $u_{rn}$  are uniquely determined by equation (2.8). This process can be easily verified by an example. The elements were determined in the following order: First row of  $R$ , second row of  $L$ ; second row of  $R$  and so forth. Another order in which they can be determined is the first row of  $R$ , first column of  $L$ ; second row of  $R$ , second column of  $L$ , and so forth.

Rutishauser (9) generalizes equations (2.7) and (2.8) for computing  $L$  and  $R$  into recursion formulas.

$$\begin{aligned} u_{ij} &= a_{ij} - \sum_{k=1}^{i-1} d_{ik} u_{kj} & \text{for } i=1,2,\dots,j \\ d_{ij} &= \frac{a_{ij} - \sum_{k=1}^{j-1} d_{ik} u_{kj}}{u_{jj}} & \text{for } i=j+1,\dots,n \end{aligned} \quad \text{for } j=1,\dots,n \quad (2.9)$$

Wilkinson (12), (13) states that if the triangular decomposition of a non-singular matrix  $A$  exists, it is unique. For if

$$A = L_1 R_1 = L_2 R_2 \quad (2.10)$$

then since

$$A = |L_1| |R_1| = |L_2| |R_2|, \quad (2.11)$$

$R_1$  and  $R_2$  are both non-singular. Therefore,

$$L_2^{-1} L_1 = R_2 R_1^{-1}. \quad (2.12)$$

The matrix  $L_2^{-1} L_1$  is unit left triangular, and the matrix  $R_2 R_1^{-1}$  is right triangular. Hence each side of (2.12) is the identity matrix and  $L_1 = L_2, R_1 = R_2$ .

There are simple matrices for which triangular decomposition breaks down. An example cited by Wilkinson (13) of a non-singular, well-conditioned matrix  $A$  which does not have a triangular decomposition is

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 1 \\ 4 & 6 & 7 \end{bmatrix} \quad (2.13)$$

Substituting in equation (2.9), the diagonal element  $u_{22} = 0$ , and the equation defining  $d_{32}$  is  $4*2 + d_{32} * 0 = 6$ . Hence,  $d_{32}$  is undefined, and the matrix (2.13) has no decomposition.

Another case is that of a matrix  $A$  whose decomposition is not unique.

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 1 \\ 3 & 3 & 1 \end{bmatrix} \quad (2.14)$$

Once more substituting in (2.9), the diagonal element  $u_{22} = 0$ , but the equation defining  $d_{32}$  is  $3*1 + d_{32} * 0 = 3$ . Thus  $d_{32}$  is arbitrary, and the matrix  $A$  of (2.14) has an infinite number of triangular decompositions. Note that the matrix in (2.14) is singular.

The relationship between triangular decomposition and Gaussian elimination is illustrated by Wilkinson (12). Gaussian elimination is an elementary procedure which may be used to reduce a matrix to right triangular form. It consists of  $(n-1)$  steps, each of which eliminate the elements of a column below the diagonal. Before the  $r^{\text{th}}$  step a non-singular matrix  $X_1$  has been reduced to the form  $X_r$  shown below for  $n = 6, r = 3$ .

$$X_3 = \begin{bmatrix} x & x & x & x & x & x \\ 0 & x & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & x & x & x & x \end{bmatrix} \quad (2.15)$$

On the  $r^{\text{th}}$  step matrix  $X_r$  is premultiplied by an elementary matrix  $M_r$ . It may be illustrated for  $n = 6$ ,  $r = 3$ .

$$M_3 = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & -m_{43} & 1 & & \\ & & -m_{53} & & 1 & \\ & & -m_{63} & & & 1 \end{bmatrix} \quad (2.16)$$

Where  $m_{ir} = X_{ir}/X_{rr}$  for  $i = r+1, \dots, n$ . In general, the premultiplication by  $M_r$  results in the subtraction of a multiple of the  $r^{\text{th}}$  row from each of the rows  $(r+1)$  to  $n$ . It is important to note that rows 1 to  $(r-1)$  are unaltered in the  $r^{\text{th}}$  step, and the zeros introduced by previous steps remain unaltered.

The matrix resulting from the final step is a right triangular matrix  $X_n$

$$M_{n-1} \dots M_2 M_1 X_1 = X_n \quad (2.17)$$

Premultiplying each side of (2.17) by the inverse of  $(M_{n-1} \dots M_2 M_1)$  gives

$$X_1 = M_1^{-1} M_2^{-1} \dots M_{n-1}^{-1} X_n. \quad (2.18)$$

The matrix  $M_r^{-1}$  differs from  $M_r$  only in the signs of the subdiagonal elements, and the matrix  $M_1^{-1} M_2^{-1} \dots M_{n-1}^{-1}$  is unit left triangular; hence, in general,

$$X_1 = M X_n \quad (2.19)$$

where  $M$  is unit left triangular and  $X_n$  is right triangular. Therefore, since the decomposition of a non-singular matrix is unique, the  $M$  and  $X_n$  of Gaussian elimination must be identical with the  $L$  and  $R$  of triangular decomposition. The importance of this relationship lies in the fact

that the failure and non-uniqueness of the decomposition occurs in the same circumstances as in the Gaussian elimination procedure without pivoting.

In Gaussian elimination, it is well-known that numerical stability is maintained by the introduction of interchanges. A modified procedure which removes instability was introduced by Wilkinson (13) for the LR algorithm.

It was shown in (2.3) that one iteration of the orthodox LR transformation to form  $A_k$  consists of premultiplying the matrix  $A_{k-1}$  by a left triangular matrix  $L_{k-1}^{-1}$  to produce a right triangular matrix  $R_{k-1}$ . The similarity transformation is then completed by post-multiplying  $R_{k-1}$  by  $L_{k-1}$ . Parlett (7) explains the modified LR transformation by replacing  $L_{k-1}$  by a matrix  $L'_{k-1}$ . This matrix  $L'_{k-1}$  is a product of elementary matrices as is  $L_{k-1}$ . However,  $L'_{k-1}$  has an interchange matrix inserted between consecutive elementary matrices and therefore, is not usually triangular.

If we use  $M_i$  from (2.16) and let the  $X_1$  and  $X_n$  of (2.17) equal  $A$  and  $R$  respectively, then

$$M_{n-1} I_{n-1, (n-1)'} \cdots M_2 I_{2,2'} M_1 I_{1,1'} A = R. \quad (2.20)$$

In the  $r^{\text{th}}$  step we premultiply by a stabilized elementary matrix  $M_r I_{r,r'}$ . The premultiplication by  $I_{r,r'}$  results in the interchange of rows  $r$  and  $r'$  where  $r' \geq r$ . The index  $r'$  is defined by  $|a_{r',r}| = \max_{i \neq r, \dots, n} |a_{ir}|$  so that the diagonal element is not exceeded in magnitude by any element below it.

This process of interchanging is usually referred to as 'partial pivoting.' A similarity transformation of  $A$  can now be completed by post-multiplying  $R$  in (2.20) by  $L'$  which may be written as

$$L' = I_{1,1}' M_1^{-1} I_{2,2}' M_2^{-1} \cdots I_{n-1,(n-1)'} M_{n-1}^{-1}.$$

This gives

$$\begin{aligned} & M_{n-1}^{-1} I_{n-1,(n-1)'} \cdots M_2^{-1} I_{2,2}' M_1^{-1} I_{1,1}' A I_{1,1}' \\ & M_1^{-1} I_{2,2}' M_2^{-1} \cdots I_{n-1,(n-1)'} M_{n-1}^{-1} \\ & = R I_{1,1}' M_1^{-1} I_{2,2}' M_2^{-1} \cdots I_{n-1,(n-1)'} M_{n-1}^{-1} \end{aligned} \quad (2.21)$$

If no interchanges are necessary so that  $I_{r,r'} = I$  for  $r = 1, 2, \dots, (n-1)$ , the product preceeding  $A$  in (2.21) would be the matrix  $L^{-1}$  of (2.16) so that  $L^{-1} A = R$  or  $A = LR$  as in (2.1). Also, the right side of (2.21) would be  $RL$ . The modified LR transformation is described by Wilkinson (13) as the reduction of the matrix  $A_k$  to a right triangular matrix  $R_k$  using Gaussian elimination with interchanges. Then  $A_{k+1}$  is formed by post-multiplying  $R_k$  by the inverse of the factors used in the reduction.

To illustrate this new factorization of the matrix  $A$ , consider the relationship between  $A_1$  and  $A_2$  for a case when  $n = 5$ . Using the left side of equation (2.21), it follows that

$$\begin{aligned} A_2 &= \begin{bmatrix} M_4 I_{4,4}' & M_3 I_{3,3}' & M_2 I_{2,2}' & M_1 I_{1,1}' \\ M_1^{-1} I_{2,2}' & M_2^{-1} I_{3,3}' & M_3^{-1} I_{4,4}' & M_4^{-1} \end{bmatrix} A_1 \begin{bmatrix} I_{1,1}' \\ I_{2,2}' \\ I_{3,3}' \\ I_{4,4}' \end{bmatrix} \\ &= \begin{bmatrix} M_4 I_{4,4}' & M_3 (I_{4,4}' I_{4,4}') I_{3,3}' & M_2 (I_{3,3}' I_{4,4}' I_{4,4}' I_{3,3}') I_{2,2}' & M_1 (I_{2,2}' I_{3,3}' I_{4,4}' I_{4,4}' I_{3,3}') I_{1,1}' \\ I_{4,4}' & I_{3,3}' & I_{2,2}' & I_{1,1}' \end{bmatrix} A_1 \times \\ & \times \begin{bmatrix} I_{1,1}' (I_{2,2}' I_{3,3}' I_{4,4}' I_{4,4}' I_{3,3}' I_{2,2}') M_1^{-1} I_{2,2}' \\ (I_{3,3}' I_{4,4}' I_{4,4}' I_{3,3}') M_2^{-1} I_{3,3}' (I_{4,4}' I_{4,4}') M_3^{-1} I_{4,4}' M_4^{-1} \end{bmatrix} \end{aligned} \quad (2.22)$$

The terms in parentheses are readily seen to be equal to I. Regrouping the parentheses in (2.22) and letting  $I_{4,4}, I_{3,3}, I_{2,2}, I_{1,1} = P^{-1}$

$$A_2 = \left[ (M_4)(I_{4,4}, M_3 I_{4,4})(I_{4,4}, I_{3,3}, M_2 I_{3,3}, I_{4,4})(I_{4,4}, I_{3,3}, I_{2,2}, M_1 I_{2,2}, I_{3,3}, I_{4,4}) P^{-1} \right] A_1 \times \quad (2.23)$$

$$\times \left[ P (I_{4,4}, I_{3,3}, I_{2,2}, M_1^{-1} I_{2,2}, I_{3,3}, I_{4,4})(I_{4,4}, I_{3,3}, M_2^{-1} I_{3,3}, I_{4,4})(I_{4,4}, M_3^{-1} I_{4,4})(M_4^{-1}) \right]$$

The matrices in the parenthesis of (2.23) and hence their product  $\tilde{L}^{-1}$  are left triangular. The modified LR algorithm using elimination with interchanges from (2.20) corresponds to

$$A_1 = P_1 \tilde{L}_1 R_1 \quad (2.24)$$

The new iterate of A is given by

$$A_2 = R_1 P_1 \tilde{L}_1 = \tilde{L}_1^{-1} P_1^{-1} A_1 P_1 \tilde{L}_1. \quad (2.25)$$

The matrix  $P_1^{-1} A_1 P_1$  is  $A_1$  with its rows and columns permuted. Since  $\tilde{L}^{-1}$  and  $\tilde{L}$  are left triangular, they are similar to  $A_k = L_{k-1}^{-1} A_{k-1} L_{k-1}$ .

Rutishauser's (9) proof of the convergence of the orthodox LR algorithm does not apply to the modified procedure. Wilkinson (13) states that if convergence to right triangular form does occur in the modified process and if none of the eigenvalues is zero, then, since the subdiagonal elements are tending to zero, the interchanges must ultimately cease. A satisfactory proof of convergence of the modified LR algorithm has not yet been published.

The LR algorithm is not suitable for full matrices because of the high volume of computations. Thus, it is necessary to reduce the original matrix to some condensed form which is invariant with respect to the LR transformation. One such form is the upper Hessenberg matrix which is an

almost triangular matrix with zeros in position  $(i,j)$  for  $i > j + 1$ .

A full matrix may be reduced to Hessenberg form in a stable manner by the use of similarity transformations.

One such method described in Wilkinson (10), (13) for reducing a matrix to this form is Gaussian elimination with pivoting on the maximum element. The original matrix  $A$  is reduced to upper Hessenberg form in  $(n-2)$  steps. The  $r^{\text{th}}$  step eliminates the elements  $(r+2, r)$  through  $(n, r)$ . Before the  $r^{\text{th}}$  step, the matrix  $A$  has been reduced to the form shown before for  $n = 6$ ,  $r = 3$ .

$$A_{r-1} = \begin{bmatrix} h_{11} & h_{12} & x & x & x & x \\ h_{12} & h_{22} & x & x & x & x \\ 0 & h_{32} & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & x & x & x & x \\ 0 & 0 & x & x & x & x \end{bmatrix} \quad (2.25)$$

where the  $h_{ij}$  elements are those of the final Hessenberg matrix  $H$ .

In the first step of the reduction, the original matrix  $A$  is premultiplied by  $M_1 I_{1,1}$ , where  $M_1$  is of the form defined in (2.16) and  $I_{1,1}$  is the interchange matrix used in (2.20). This eliminates the  $(3,1)$  to  $(n,1)$  elements. The similarity transformation is completed by multiplying on the right by  $I_{1,1}' M_1^{-1}$  to produce the matrix

$$M_1 I_{1,1}' A I_{1,1} M_1^{-1}.$$

Continue in this way for  $(n-2)$  steps to produce the Hessenberg matrix.

$$M_{n-2} I_{n-2,(n-2)}' \cdots M_2 I_{2,2}' M_1 I_{1,1}' A I_{1,1} M_1^{-1} \quad (2.26)$$

$$I_{2,2}' M_2^{-1} \cdots I_{n-2,(n-2)}' M_{n-2}^{-1} = H.$$



Wilkinson (11), (13) develops another method for the reduction of the original matrix. It is Householder's method which involves a similarity transformation using an elementary orthogonal matrix. Again, there are  $(n-2)$  steps in this reduction. In the  $r^{\text{th}}$  step, zeros are introduced in the  $r^{\text{th}}$  column without destroying the zeros introduced in the previous  $(r-1)$  steps. Immediately before the  $r^{\text{th}}$  step, matrix  $A$  has been reduced to a form illustrated in (2.25).

An orthogonal matrix  $P$  is defined by  $P = I - 2ww^T$  where  $w$  is a unit vector such that  $w^T w = 1$ . The vector  $w$  has  $n$  components, the first  $r$  of which are equal to zero. Then

$$P_r = I - 2 w_r w_r^T = I - u_r u_r^T / 2K_r^2 \quad (2.27)$$

where

$$u_{ir} = 0 \quad \text{for } i = 1, 2, \dots, r \quad (2.28)$$

$$u_{r+1,r} = a_{r+1,r} / S_r, \quad u_{ir} = a_{ir} \quad \text{for } i = r+2, \dots, n$$

$$S_r = \left( \sum_{i=r+1}^n a_{ir}^2 \right)^{1/2}, \quad 2K_r^2 = S_r^2 + a_{r+1,r}^2$$

For  $P$  to be as accurately orthogonal as possible, the sign of  $S$  in the equation defining  $u_{r+1,r}$  and  $2K_r^2$  must be taken as that of  $a_{r+1,r}$ . The new value of  $a_{r+1,r}$  is computed directly to be  $\pm S$ .

The matrix  $A_r$  is defined by the relation

$$A_r = P_r^T A_{r-1} P_r \quad \text{for } r = 2, \dots, n-1. \quad (2.29)$$

$A_r$  is premultiplied by  $P_r^T$  such that

$$P_r^T A_{r-1} = (I - u_r u_r^T / 2K_r^2) A_{r-1} = A_{r-1} - u_r (u_r^T A_{r-1}) / 2K_r^2 = F_r \quad (2.30)$$

and post-multiplied by  $P_r$ .

$$\begin{aligned} A_r &= P_r^T A_{r-1} P_r = F_r P_r = F_r (I - u_r u_r^T / 2K_r^2) \\ &= F_r - (F_r u_r) u_r^T / 2K_r^2 \end{aligned} \quad (2.31)$$

If

$$u_r^T A_{r-1} = p_r^T \quad (2.32)$$

where  $p_r^T$  has its first  $r$  elements equal to zero because of the zeros in  $u_r$  and  $A_{r-1}$ , then

$$F_r = A_{r-1} - (u_r / 2K_r^2) p_r^T \quad (2.33)$$

For the post-multiplication of  $q_r$  is defined by

$$F_r u_r = q_r. \quad (2.34)$$

The vector  $q_r$  has no zero components. Finally,

$$A_r = F_r - q_r (u_r / 2K_r^2)^T. \quad (2.35)$$

After  $(n-2)$  steps of (2.29) the original matrix  $A$  is reduced to upper Hessenberg form.

The convergence to zero of the subdiagonal elements  $a_{ij}^{(k)}$  ( $i > j$ ) of  $A_k$  is usually determined by the quantities  $(|\lambda_i| / |\lambda_j|)^k$  as  $k \rightarrow \infty$ . In the case of a Hessenberg matrix, the only subdiagonal elements are the  $a_{r+1,r}^{(k)}$ ,  $r = 2, \dots, n$ . Convergence is linear and depends on the ratios  $|\lambda_i| : |\lambda_j|$ . If  $|\lambda_i / \lambda_j|$  is close to unity, convergence is slow. Slow convergence can easily be avoided by a simple acceleration technique involving shifts.

A scheme for shifting the origin was first suggested by Rutishauser (9) and later improved by Wilkinson (13). Let  $p_k$  be an approximation to  $\lambda_n$  where  $\lambda_n$  is the eigenvalue appearing in the  $(n,n)$  position of  $A_k$ . The LR transformation is applied to  $(A_k - p_k I)$  rather than to  $A_k$ . This matrix has eigenvalues  $(\lambda_i - p_k)$ . According to Parlett (7),  $a_{n,n-1}$  tends to zero as  $\{(\lambda_n - p_k) / (\lambda_{n-1} - p_k)\}^k$ .

The transformation produces a sequence of matrices defined by

$$A_k - p_k I = L_k R_k$$

and

$$R_k L_k + p_k I = A_{k+1} \quad (.36)$$

or alternatively

$$\begin{aligned} A_{k+1} &= R_k L_k + p_k I = L_k^{-1} (A_k - p_k I) L_k + p_k I = \\ &= L_k^{-1} A_k L_k. \end{aligned} \quad (2.37)$$

and the matrices are still similar to  $A_1$ .

In fact, from (2.37)

$$\begin{aligned} A_{k+1} &= L_k^{-1} A_k L_k = L_k^{-1} L_{k-1}^{-1} A_{k-1} L_{k-1} L_k = L_k^{-1} \dots \\ &= L_2^{-1} L_1^{-1} A_1 L_1 L_2 \dots L_k \end{aligned} \quad (2.38)$$

or

$$L_1 L_2 \dots L_k A_{k+1} = A_1 L_1 L_2 \dots L_k$$

According to Wilkinson (13) this modification is described as the LR with shifts of origin and 'restoring' because the shift is added back at each stage. There is sometimes an advantage to using the 'non-restoring' process defined by

$$\begin{aligned} A_k &= p_k I = L_k R_k \\ R_k L_k &= A_{k+1} \end{aligned} \quad (2.39)$$

so that

$$\begin{aligned} A_{k+1} &= R_k L_k = L_k^{-1} (A_k - p_k I) L_k = L_k^{-1} A_k L_k - p_k I \\ &= L_k^{-1} (L_{k-1}^{-1} A_{k-1} L_{k-1} - p_{k-1} I) L_k - p_k I \\ &= L_k^{-1} L_{k-1}^{-1} A_{k-1} L_{k-1} L_k - (p_{k-1} + p_k) I \end{aligned} \quad (2.40)$$

Continuing this argument as in (2.38)

$$\begin{aligned} A_{k+1} &= L_k^{-1} (A_k - p_k I) L_k = L_k^{-1} \dots L_1^{-1} A_1 L_1 \dots \\ &= L_k^{-1} - (p_1 + p_2 + \dots + p_k) I \\ &= L_k^{-1} \dots L_1^{-1} [A_1 - (p_1 + p_2 + \dots + p_k) I] L_1 \dots L_k. \end{aligned} \quad (2.41)$$

By using this technique, it may be observed that the eigenvalues of  $A_{k+1}$  differ from those of  $A_1$  by  $\sum_{i=1}^k p_i$ .

Now, returning to the restoring process and showing that equation (2.6) holds for this modification

$$\begin{aligned}
 & L_1 L_2 \dots L_{k-1} (L_k R_k) R_{k-1} \dots R_2 R_1 \\
 &= L_1 L_2 \dots L_{k-1} (A_k - p_k I) R_{k-1} \dots R_2 R_1 \\
 &= (A_1 - p_k I) L_1 L_2 \dots L_{k-1} R_{k-1} \dots R_2 R_1 \quad (2.42) \\
 &= (A_1 - p_k I) (A_1 - p_{k-1} I) L_1 L_2 \dots L_{k-2} R_{k-2} \dots R_2 R_1 \\
 &= (A_1 - p_k I) (A_1 - p_{k-1} I) \dots (A_1 - p_1 I)
 \end{aligned}$$

Hence, using  $T_k$  and  $U_k$  defined as in (2.5),  $T_k U_k$  gives the triangular decomposition of  $\prod_{i=1}^k (A_1 - p_i I)$ .

A frequent choice of  $p_k$  is an eigenvalue of the  $2 \times 2$  principal submatrix

$$\begin{bmatrix} a_{n-1,n-1} & a_{n-1,n} \\ a_{n,n-1} & a_{n,n} \end{bmatrix} \quad (2.43)$$

which is closest to  $a_{n,n}$ . Ralston (8) states that the importance of choosing  $p_k$  in this manner is that it replaces the iteration's linear convergence  $(|\lambda_n|/|\lambda_{n-1}|)^k$  with quadratic convergence  $(|\lambda_n|/|\lambda_{n-1}|)^{2k}$ .

Several other methods of choosing  $p_k$  will be discussed in Chapter III.

This method of origin shifting accelerates the convergence of the last diagonal element of  $A$ . When the remaining elements of row  $n$  are effectively zero, the element  $a_{nn}^{(k)}$  will be a good approximation to  $\lambda_n$ .

The QR transformation proposed by Francis (2), (3) uses a decomposition of an arbitrary matrix  $A$  into the product of a unitary matrix  $Q$  and a right triangular matrix  $R$ . A sequence of matrices can be defined

starting with  $A = A_1$  such that

$$A_k = Q_k R_k \quad (2.44)$$

$$A_{k+1} = R_k Q_k \quad \text{for } k = 1, 2, \dots, n.$$

$A_{k+1}$  is formed by post-multiplying  $R_k$  by  $Q_k$ . This algorithm can also be written as a similarity transformation.

$$\begin{aligned} A_k &= R_{k-1} Q_{k-1} = Q_{k-1}^H A_{k-1} Q_{k-1} \\ &= Q_{k-1}^H \dots Q_2^H Q_1^H A_1 Q_1 Q_2 \dots Q_{k-1}. \end{aligned} \quad (2.45)$$

Equation (2.45) gives

$$Q_1 Q_2 \dots Q_{k-1} A_k = A_1 Q_1 Q_2 \dots Q_{k-1}. \quad (2.46)$$

Francis (2) develops a fundamental result similar to that which

Rutishauser (9) developed for the LR transformation. Using  $A_k = Q_k R_k$  and equation (2.46), it follows that

$$\begin{aligned} Q_1 Q_2 \dots Q_k R_k &= A_1 Q_1 Q_2 \dots Q_{k-1} \\ Q_1 Q_2 \dots Q_{k-1} R_{k-1} &= A_1 Q_1 Q_2 \dots Q_{k-2} \end{aligned}$$

and soon.

If  $P_k = Q_1 Q_2 \dots Q_k$  and  $S_k = R_k R_{k-1} \dots R_1$ , where  $P_k$  is unitary and  $S_k$  is right triangular, then

$$\begin{aligned} P_k S_k &= Q_1 Q_2 \dots Q_k R_k R_{k-1} \dots R_1 \\ &= Q_1 Q_2 \dots Q_{k-1} A_k R_{k-1} \dots R_2 R_1 \\ &= A_1 Q_1 Q_2 \dots Q_{k-1} R_{k-1} \dots R_2 R_1 \\ &= A_1^2 Q_1 Q_2 \dots Q_{k-2} R_{k-2} \dots R_2 R_1 \quad \text{from (2.46)} \\ &= A_1^k \end{aligned} \quad (2.47)$$

Francis (2) proved that for any matrix  $A$  there exists a unitary matrix  $Q$  such that  $A = QR$  where  $R$  is a right triangular matrix which has real non-negative, diagonal elements. Moreover, the  $Q$  is unique if  $A$  is non-singular. Thus the unitary-triangular decomposition of any square matrix exists, and, if the matrix is non-singular, the decomposition is

unique. There is no possibility for breakdown in the decomposition as in the QR transformation.

Francis (2), (3) proved that if  $A$  is a non-singular matrix with all eigenvalues of distinct moduli, then, as  $k \rightarrow \infty$ , the elements of  $A_k$  below the diagonal tend to zero, and the elements on the diagonal tend to the eigenvalues of  $A$ .

The factorization of a matrix  $A_k$  into  $Q_k$  and  $R_k$  involves the use of elementary unitary transformations. Instead of determining the matrix  $Q_k$  directly, we find the matrix  $Q_k^H$  such that

$$Q_k^H A_k = R_k \quad (2.48)$$

where  $Q_k^H = \overline{Q_k}^T$ . The matrix is determined in factorized form either as the product of plane rotations using the Givens triangularization or as the product of elementary Hermitian matrices using the Householder triangularization.

The unitary matrix used in Givens' process differs from the identity matrix by a submatrix. The matrix is defined by Francis (3) as

$$\begin{aligned} t_{ii} &= e^{i\alpha} \cos \theta & t_{ij} &= -e^{i\beta} \sin \theta \\ t_{ji} &= e^{i\gamma} \sin \theta & t_{jj} &= e^{i\delta} \cos \theta \end{aligned} \quad (2.49)$$

where  $\theta, \alpha, \beta, \gamma, \delta$  are real and  $\alpha - \beta - \gamma + \delta = 0 \pmod{2\pi}$ .

Since this thesis is concerned with real matrices, we will develop the remainder of this process in terms of orthogonal instead of unitary transformations. For real matrices  $\alpha = \beta = \gamma = \delta = 0$ . Therefore, the orthogonal matrix  $T_{ij}$  is defined by Faddeev and Faddeeva (2) as

$$T_{ij} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & C & \dots & -S \\ & & & \vdots & \ddots & \vdots \\ & & & & 1 & \\ & & S & \dots & \dots & C \end{bmatrix} \begin{matrix} \dots i \\ \\ \\ \dots j \end{matrix} \quad (i < j) \quad (2.50)$$

Where  $C = \cos \theta$  and  $S = \sin \theta$ . This plane rotation matrix  $T_{ij}$  corresponds to a rotation in the  $(i, j)$  plane, and the angle  $\theta$  of the rotation is chosen so as to reduce the  $(i, j)$  element of  $A_k$  to zero.

Equation (2.48) can be rewritten for real matrices.

$$Q_k^T A_k = R_k \quad (2.51)$$

The elements of  $A_k$  below the diagonal in the first column are eliminated one at a time with a  $T_{ij}$  matrix. This process is repeated on the first  $(n-1)$  columns until matrix  $A_k$  is reduced to the right triangular matrix  $R_k$ .  $Q_k^T$  is the product of these plane rotations. From Ralston (8)

$$R_k = \left( \prod_{i=1}^{n-1} \prod_{j=i+1}^n S_{ji} \right) A_k = Q_k^T A_k \quad (2.52)$$

The matrix  $R_k Q_k$  is then computed by successive post-multiplication of  $R_k$  with the transposes of the factors of  $Q_k^T$ .

For a full matrix Given's method explained above requires considerably more work than Householder's method. The QR transformation is practical only when applied to a matrix of Hessenberg form. The reason is the same as that stated previously for the LR transformation. The almost triangular form of a matrix is preserved under the QR transformation because the matrices  $Q_k$  in the algorithm are also almost triangular.

The technique of shifts of origin with or without restoring may be introduced in the QR algorithm just as in the LR algorithm. For the

process with restoring, instead of  $A_k$  consider

$$\begin{aligned} A_k - p_k I &= Q_k R_k \\ R_k Q_k + p_k I &= A_{k+1} \end{aligned} \quad (2.53)$$

Continuing

$$\begin{aligned} A_{k+1} &= R_k Q_k + p_k I = Q_k^T (A_k - p_k I) Q_k + p_k I = \\ &= Q_k^T A_k Q_k \end{aligned} \quad (2.54)$$

Just as in (2.38)

$$A_{k+1} = Q_k^T A_k Q_k = (Q_1 Q_2 \dots Q_k)^T A (Q_1 Q_2 \dots Q_k) \quad (2.55)$$

Francis (3) shows that (2.47) holds for the QR since

$$\begin{aligned} Q_1 Q_2 \dots Q_{k-1} (Q_k R_k) R_{k-1} \dots R_2 R_1 \\ &= Q_1 \dots Q_{k-1} (A_1 - p_k I) R_{k-1} \dots R_1 \\ &= (A_1 - p_k I) Q_1 \dots Q_{k-1} R_{k-1} \dots R_1 \\ &= (A_1 - p_k I) (A_1 - p_{k-1} I) Q_1 \dots Q_{k-2} R_{k-2} \dots R_1 \\ &= (A_1 - p_k I) (A_1 - p_{k-1} I) \dots (A_1 - p_1 I) \end{aligned} \quad (2.56)$$

Hence, using  $P_k$  and  $S_k$  as previously defined,  $P_k S_k$  gives the decomposition of  $\prod_{i=1}^k (A_1 - p_i I)$ . The shifts  $p_k$  are chosen as in the LR transformation.

The QR algorithm just defined is not very satisfactory when the real matrix being considered has complex eigenvalues. In this case the origin shifts will be complex at some stage, and the matrix  $A_{k+1}$  will be complex. Now, if any iteration with a complex shift  $p_k$  is followed by one using the conjugate shift, so that  $p_{k+1} = \bar{p}_k$ , then the matrix  $A_{k+2}$  will be real. Therefore, at the expense of doing two complex iterations instead of one when a matrix has a complex pair of eigenvalues, the resulting matrix is real and the convergence is still accelerated.



Francis (2), (3) develops an algorithm which produces a real matrix from which both complex and real eigenvalues can be calculated. This algorithm does not involve complex arithmetic in the intermediate steps. Consider the effect of performing two steps of the QR. From (2.45) it follows that

$$A_3 = Q_2^T A_2 Q_2 = Q_2^T Q_1^T A_1 Q_1 Q_2 \quad (2.57)$$

and

$$\begin{aligned} Q_1 Q_2 R_2 R_1 &= Q_1 (A_2 - p_2 I) R_1 \\ &= (A_1 - p_2 I) Q_1 R_1 \\ &= (A_1 - p_2 I)(A_1 - p_1 I) \end{aligned} \quad (2.58)$$

The matrix on the right of (2.58) is real. If  $Q_1 Q_2 = Q_0$  and  $R_2 R_1 = R_0$ , then from (2.57)  $A_3 = Q_0^T A_1 Q_0$ , and by uniqueness

$$Q_0 R_0 = (A_1 - p_1 I)(A_1 - p_2 I). \quad (2.59)$$

From (2.59) it follows that

$$Q_0^T (A_1 - p_1 I)(A_1 - p_2 I) = R_0, \quad (2.60)$$

and hence  $Q_0^T$  is the real orthogonal matrix that reduced  $(A_1 - p_1 I)(A_1 - p_2 I)$  to right triangular form.

An explanation of a property of Hessenberg matrices is necessary and is given by Parlett (7). If a matrix  $M$  is unitarily similar to an upper Hessenberg matrix  $H$  with real, non-negative, subdiagonal elements, then the first column of the transformation matrix  $U$  uniquely determines the remaining columns of  $U$  and all of  $H$ . To make this clear, consider

$$UH = MU \quad (2.61)$$

and develop recursive formulas for calculating the  $U_i$  and  $h_{ij}$  from  $U_1$  and  $M$ . At the  $j^{\text{th}}$  stage, we take  $H_1, \dots, H_{j-1}$  and  $U_1, \dots, U_j$  as known where  $j = 1, 2, \dots, n$  and subscripts on matrices denoting columns.

Equate the  $j^{\text{th}}$  column of each side of (2.61) and obtain the formulas for  $H_j$  and  $U_{j+1}$ . Since  $U_1$  is orthogonal to the other columns of  $U_j$ ,

$$h_{ij} = U_i^T MU_j \quad \text{for } i = 1, 2, \dots, j \quad (2.62)$$

Take all known quantities of (2.62) to the right side of  $UH_j = MU_j$ . Then

$$h_{j+1,j} U_{j+1} = MU_j - \sum_{i=1}^j h_{ij} U_i = V_{j+1}. \quad (2.63)$$

Since  $U_{j+1}$  must be of unit length, it follows that

$$h_{j+1,j} = \|V_{j+1}\|, \quad (2.64)$$

$$U_{j+1} = \left( \frac{1}{h_{j,j+1}} \right) V_{j+1}$$

Although this procedure ends when the first  $h_{j+1,j} = 0$ , Francis (3) chooses  $U_1$  so that it determines the entire transformation.

Using this property, Francis (2), (3) developed the double QR algorithm which must be applied to a Hessenberg matrix. The orthogonal transformation of  $A_1$  into  $A_3$  using  $Q_0$  from equation (2.59) is uniquely determined by the first column of  $Q_0$ .  $R_0$  is right triangular, and, therefore, column 1 of  $Q_0$  is column 1 of  $Q_0 R_0$  normalized to unit length. Only the first column of  $Q_0 R_0$  is needed to compute  $A_3$ . This column consists of at most three elements which are given by

$$X_1 = (a_{11} - p_1)(a_{11} - p_2) + a_{12} a_{21} = a_{11}^2 + a_{12} a_{21} - a_{11} (p_1 + p_2) + p_1 p_2 \quad (2.65)$$

$$Y_1 = a_{21} (a_{11} - p_2) + (a_{22} + p_1) a_{21} = a_{21} (a_{11} + a_{22} - p_1 - p_2)$$

$$Z_1 = a_{32} a_{21}$$

To start the iteration, apply to  $A_1$  an orthogonal matrix, say  $P_1$  which has as its first column the elements of (2.65). This destroys the Hessenberg form by changing the first three rows and columns of  $A_k$ . Thus,

$$A_k = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \cdot & \cdot & a_{1n} \\ a_{12} & a_{22} & a_{23} & a_{24} & \cdot & \cdot & a_{2n} \\ & a_{23} & a_{33} & a_{34} & \cdot & \cdot & a_{3n} \\ & & a_{43} & a_{44} & \cdot & \cdot & a_{4n} \\ & & & a_{54} & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \cdot \\ & & & & & \cdot & a_{nn} \end{bmatrix} \quad (2.66)$$

$$\rightarrow \begin{bmatrix} \underline{a_{11}}' & \underline{a_{12}}' & \underline{a_{13}}' & \underline{a_{14}}' & \cdot & \cdot & \underline{a_{1n}} \\ \underline{a_{21}}' & \underline{a_{22}}' & \underline{a_{23}}' & \underline{a_{24}}' & \cdot & \cdot & \underline{a_{2n}} \\ \underline{a_{31}}' & \underline{a_{32}}' & \underline{a_{33}}' & \underline{a_{34}}' & \cdot & \cdot & \underline{a_{3n}} \\ \underline{a_{41}}' & \underline{a_{42}}' & \underline{a_{43}}' & \underline{a_{44}}' & \cdot & \cdot & \underline{a_{4n}} \\ & & & a_{54} & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \cdot \\ & & & & & \cdot & a_{nn} \end{bmatrix} = P_1^T A_1 P_1$$

where the elements changed by the row and column operations are underlined and primed, respectively. Next, the matrix  $P_1^T A_1 P_1$  is restored to upper Hessenberg form with non-negative subdiagonal elements using any orthogonal procedure. This reduction is accomplished by applying to  $P_1^T A_1 P_1$ , transformations  $P_2, \dots, P_{n-2}$  such that

$$P_{n-2}^T \dots P_1^T A_1 P_1 \dots P_{n-2} = A_3. \quad (2.67)$$

It is necessary that the first column of  $P_1 P_2 \dots P_{n-2}$  be the first column of  $P_1$ .

A typical stage in the iteration is illustrated by

$$\begin{bmatrix}
 h & h & h & z & z & z & z & z \\
 & h & h & h & z & z & z & z \\
 & & h & h & z & z & z & z \\
 & & & z & z & z & z & z \\
 & & & z & z & z & z & z \\
 & & & z & z & z & a & a \\
 & & & & & & a & a \\
 & & & & & & & a
 \end{bmatrix} \rightarrow \begin{bmatrix}
 h & h & h & h' & z' & z & z & z \\
 & h & h & h & h' & z' & z' & z \\
 & & h & h & h' & z' & z' & z \\
 & & & \underline{h} & \underline{h'} & \underline{z'} & \underline{z'} & \underline{z} \\
 & & & \underline{0} & \underline{z'} & \underline{z'} & \underline{z'} & \underline{z} \\
 & & & \underline{0} & \underline{z'} & \underline{z'} & \underline{z'} & \underline{z} \\
 & & & & z' & z' & z' & a \\
 & & & & & & & a
 \end{bmatrix}$$

where the elements of the three rows and columns changed by the iteration are indicated as in (2.66). The elements  $a$  and  $h$  are those of the initial and final matrices  $A_k$  and  $A_{k+2}$ , respectively. The matrix  $A_{k+2}$  has upper Hessenberg form.

For the double QR algorithm, Householder's method is more economical than Given's. Householder's procedure, using floating-point arithmetic, was previously explained for the reduction of a matrix to Hessenberg form. It is used in fixed-point arithmetic for the double QR algorithm. The equations for this version are given by Wilkinson (13) to be

$$\begin{aligned}
 P &= I - 2 p_r p_r^T / \| p_r \|^2 \\
 p_r^T &= (0, \dots, 0, 1, u_r, v_r, 0, \dots, 0) \\
 s_r^2 &= x_r^2 + y_r^2 + z_r^2, \quad \bar{u}_r = y_r / (x_r \mp s_r), \quad \bar{v}_r = z_r / (x_r \mp s_r) \\
 2 / \| p_r \|^2 &= 2 / (1 + u_r^2 + v_r^2).
 \end{aligned} \tag{2.68}$$

The elements  $a_{r,r-1}^{(r-1)}$ ,  $a_{r+1,r-1}^{(r-1)}$ , and  $a_{r+2,r-1}^{(r-1)}$  are denoted by  $x_r$ ,  $y_r$ ,  $z_r$ .

With this choice of  $P$ , orthogonal similarity transformations are performed on the matrix in (2.66) to return it to upper Hessenberg form. This double iteration technique is much preferred to the original QR algorithm with a complex matrix in the intermediate step.

### III. DISCUSSION AND COMPARISON OF ALGORITHMS

Several difficulties are encountered when attempting to find eigenvalues by the orthodox LR algorithm. They are summed up in the following:

1. The  $\frac{2}{3} n^3$  multiplications required for one iteration on a full matrix is prohibitive.
2. As illustrated in (2.13) and (2.14), matrices exist for which triangular decomposition fails or is not unique.
3. A more frequent problem occurs when the triangular decomposition exists but is numerically unstable.
4. The convergence of the subdiagonal elements to zero is often very slow if no origin shifts are used.

If any matrix is even close to a matrix which does not have a triangular decomposition, although the LR algorithm does not actually break down, it becomes numerically unstable. This occurs when some of the divisors in the triangular decomposition are small, causing the error to be magnified. When the divisors are zero, the decomposition breaks down. This complete breakdown is rare, but numerical instability of the orthodox LR algorithm is quite common. For this method to be useful, it must be stable with respect to the buildup of round off error. So the modified LR algorithm, which has less satisfactory convergence properties, must be used. The work involved in each iteration of the modified LR algorithm on a matrix of Hessenberg form is proportional to  $n^2$  which is the same as for one iteration of the orthodox LR algorithm.

To evaluate the LR and QR methods for computing the eigenvalues of a real non-symmetric matrix, programs were written in Fortran II and run on the IBM 1620 Model II.

Two programs were written for the modified LR transformation. The original matrix was reduced to upper Hessenberg form by Gaussian elimination in one and Householder's method in the other. Then origin shifts with restoring were used at every iteration to accelerate the convergence of the last diagonal element of the matrix  $A$ . The shift was chosen to be the eigenvalue of the lower right  $2 \times 2$  principal submatrix, which is closest to  $a_{nn}^{(k)}$  as was described by Wilkinson (13). If these eigenvalues were real and equal to  $s_k$  and  $t_k$ , the shift  $p_k$  was set equal to  $s_k$  or  $t_k$ , depending on the quantities  $|s_k - a_{nn}^{(k)}|$  and  $|t_k - a_{nn}^{(k)}|$ . If  $p_k$  is a good approximation to  $\lambda_n$  the element  $a_{n,n-1}^{(k)}$  decreases rapidly, and, therefore, it is advantageous to iterate on  $(A_k - p_k I)$ . When the  $a_{n,n-1}$  element is zero to working accuracy, the element  $a_{nn}$  will be a good approximation to  $\lambda_n$ . The matrix is then deflated by omitting row and column  $n$ , and the LR transformation may be applied to the reduced matrix. In this way, the eigenvalues of the matrix  $A$  were found one by one as the order of the matrix is reduced at each step.

Another necessary step is checking the subdiagonal elements. These elements are tending to zero, but, if a matrix  $A_k$  has a subdiagonal element  $a_{i+1,i}$  which is small enough, the element is set equal to zero, and the matrix is partitioned. Then continue iterating on the submatrix of order  $(n-i)$  in the bottom right hand corner.

This technique is combined with one originated by Francis (3) which also involves subdiagonal elements. When a matrix  $A_k$  is found to have

small elements  $a_{i+1,i}^{(k)}$  and  $a_{i+2,i+1}^{(k)}$ , the matrix is partitioned. This condition is determined by using the equation

$$a_{i+1,i}^{(k)} a_{i+2,i+1}^{(k)} / (a_{i+1,i+1}^{(k)} - p_k) < \delta \text{ for } i = 1, 2, \dots, n-1, \quad (3.1)$$

where  $\delta$  is some criteria for convergence and  $(a_{i+1,i+1}^{(k)} - p_k)$  is the current value of  $a_{i+1,i+1}$  after origin shifting. If the largest value for which this is true is set equal to some  $s$ , then the modified LR transformation is performed on matrix  $(A_k - p_k I)$  starting with row  $(s+1)$ . The similarity transformation is completed starting with column  $(s+1)$ . A substantial amount of computation is saved by partitioning matrix  $A_k$  in the two ways described above. Wilkinson (13) points out that the reduction to Hessenberg form is unique only if all the subdiagonal elements are non-zero. This is another reason for the importance of iterating only on the lower submatrix when a subdiagonal element is less than some  $\delta$ .

Another way to avoid instability is to make use of elementary orthogonal transformation. Two programs were written for the double QR algorithm. The original matrix was reduced to upper Hessenberg form by Gaussian elimination and Householder's method just as for the modified LR. The double QR algorithm was developed especially for the case of a real matrix which has complex eigenvalues. Before each iteration the roots of the  $2 \times 2$  principal submatrix of (2.43) are found, and the subdiagonal elements  $a_{n,n-1}^{(k)}$  and  $a_{n-1,n-2}^{(k)}$  are inspected. If either or both of these are effectively zero, the appropriate eigenvalue or eigenvalues are detected, and the matrix is reduced by one or two. If these subdiagonal elements are non-zero, iteration is performed using the double QR algorithm. This shift was incorporated at every stage.

The roots mentioned in the preceeding paragraph are the origin shifts  $p_k$  and  $p_{k+1}$ . If these roots are complex conjugates, the sum and product of  $p_k$  and  $p_{k+1}$  are used instead of using the origin shifts directly. Thus, any complex arithmetic is avoided. The complex eigenvalues of  $A$  will be the roots of the lower right principal submatrix at some stage when  $a_{n-1,n-2}^{(k)}$  is effectively zero.

This transformation leads to  $A_{(k+2)}$  only when it is unique. So, if some  $a_{i+1,i} = 0$ , it is necessary to ensure uniqueness by partitioning the matrix and iterating on only a principal submatrix. The subdiagonal elements are scanned before each iteration as was done for the modified LR algorithm, and the smallest  $q$  is found such that for  $q < i < n$  all  $a_{i+1,i} = 0$ .

The transformation is also in danger of being ill-determined because of the smallness of the subdiagonal elements. Thus, we again incorporate a device used in the modified LR which saves a significant number of calculations. When the elements  $a_{i+1,i}$  and  $a_{i+2,i+1}$  are small enough to be treated as zero, the iteration is started at  $a_{ii}$  rather than  $a_{qq}$ . A satisfactory criterion for this is the size of

$$\frac{|a_{i+1,i} a_{i+2,i+1} (|a_{i+1,i+1} + a_{i+2,i+2} - (p_k + p_{k+1})| + |a_{i+3,i+2}|) / (a_{i+1,i+1} (a_{i+1,i+1} - (p_k + p_{k+1})) + a_{i+1,i+2} a_{i+2,i+1} + p_k p_{k+1})|}{(3.2)}$$

The computer was programmed to find the largest values for which (3.2) is less than some  $\delta$ . It may be noted that  $s \leq i \leq n-3$  and  $1 \leq q \leq s < n$ . Using this system, an iteration is started at the  $s^{\text{th}}$  row and column of the matrix, and each column operation is started at the  $q^{\text{th}}$  element of each column. If  $s > q$ , it is necessary to change the sign of the element  $a_{i,i-1}^{(k)}$ .



On the first step of an iteration an orthogonal transformation was performed on  $A_1$  using the matrix  $P_1$  whose first column consists of the elements  $X_1, Y_1, Z_1$  given in (2.65). In following steps that reduce the matrix to Hessenberg form, the elements  $a_{r,r-1}^{(k-1)}, a_{r+1,r-1}^{(k-1)}$ , and  $a_{r+2,r-1}^{(k-1)}$ , which are used to determine  $u_r$  and  $v_r$  of (2.68), are denoted by  $X_r, Y_r, Z_r$ . The preliminary transformation of  $A_1, P_1^T A_1 P_1$  can be made exactly analogous to the operations used to reduce the matrix if we augment the matrix  $A_1$  on the left with the elements  $X_1, Y_1, Z_1$ . There are  $4n^2$  multiplications required by each iteration of the QR algorithm as compared to  $5n^2$  involved in passing from  $A_k$  to  $A_{(k+2)}$  using the double QR algorithm just discussed.

Several different shift strategies involving the element  $a_{nn}^{(k)}$  were incorporated in the modified LR programs for experimentation which included the following suggested by Wilkinson (13):

1. Use the shift  $a_{nn}^{(k)}$  at every stage.
2. Use the shift only when  $|1 - a_{nn}^{(k)} / a_{nn}^{(k-1)}|$  is less than some tolerance.
3. Use the shift only when  $|a_{n,n-1}^{(k)}| \leq \epsilon \|A_1\|_\infty$  where  $\epsilon$  is some tolerance.

These procedures accelerated the convergence of the eigenvalues; but, if at some stage the element  $a_{nn}$  becomes zero, the shift has no effect. However, these shifts of origin determined by  $a_{nn}^{(k)}$  are of limited value since they can obviously only be used on a real matrix known to have real eigenvalues.

A technique suggested by Francis (3) for the double QR algorithm uses the roots of the matrix in (2.43). These are denoted by  $\lambda_k$  and  $\lambda_{k+1}$  ordered to differ least from  $a_{n-1,n-1}$  and  $a_{nn}$ , respectively. We retain the two previous roots  $\lambda_{(k-2)}$  and  $\lambda_{(k-1)}$  and calculate  $\left| \frac{\lambda_k - \lambda_{k+2}}{\lambda_k} \right|$  and  $\left| \frac{\lambda_{k+1} - \lambda_{k-1}}{\lambda_{k+1}} \right|$ . If these quantities are both greater than  $1/2$ , origin shifts  $p_k$  and  $p_{k+1}$  are set, each equal to zero; if they are both less than  $1/2$ , set  $p_k = \lambda_k$  and  $p_{k+1} = \lambda_{k+1}$ . Otherwise both  $p_k$  and  $p_{k+1}$  are set equal to the real part of either  $\lambda_k$  or  $\lambda_{k+1}$ , whichever corresponds to the quantity less than  $1/2$ . This criterion of  $1/2$  is arbitrary.

A variation of this shifting technique was suggested by Wilkinson (13). For the modified LR algorithm, he uses  $|p_k / p_{k-1} - 1| < 1/2$  as a criterion. This procedure, and the one explained in the preceding paragraph, uses a shift only after the shift shows an indication of convergence. This technique resulted in a high rate of convergence in those examples tested. However, there was no significant difference in the results when the shift was used at every stage. So, for calculating the results shown in the tables in Appendix B, it was decided to use the shift at every iteration. Shifting is a technique for keeping the number of iterations at a minimum and is desirable not only from the standpoint of a decrease in computation time, but also from the standpoint of increased accuracy.

All the calculations were performed using double precision arithmetic, retaining 16 significant digits. The necessity of this high precision is questionable when performing the iterations using the double QR or modified LR transformations. However, it is important in the preliminary reduction of the general matrix to upper Hessenberg form to prevent round off and ensure the similarity of the condensed matrix to the original

matrix. The criterion for convergence was varied using  $\epsilon = 10^{-4}$ ,  $\epsilon = 10^{-6}$  and  $\epsilon = 10^{-8}$ .

One difficulty that was encountered is the sensitivity of eigenvalues. A problem is said to be ill-conditioned if small changes in the parameters make comparatively large changes in the solutions. In this case, the parameters are the elements of the matrix. It is meaningless to state that a matrix  $A$  is ill-conditioned. It may be ill-conditioned with respect to the calculation of the inverse or the calculation of eigenvalues or eigenvectors, but ill-conditioning with respect to one does not necessarily imply ill-conditioning with respect to another. Since only the eigenvalues are being discussed in this thesis, the sole concern is with the sensitivity of the eigenvalues to changes in the parameters which determine the condition of a problem. Due to the sensitivity of some eigenvalues, it is useless to seek a single computer method which is accurate in all cases.

It was mentioned in the preceeding chapter that after the matrix has become triangular when using the orthodox LR algorithm, the eigenvalues appear on the diagonal in decreasing order of magnitude from left to right. This condition is considered desirable but not necessary. Application of the modified LR algorithm to a matrix frequently causes the eigenvalues to become disordered along the diagonal. This is nearly always the case when the disappearance or smallness of subdiagonal elements necessitates partitioning the matrix. Applying these strategies to the double QR algorithm also makes it less likely that the eigenvalues will be found in order of decreasing magnitude.

Another case that must be considered is that of a matrix having some eigenvalues of equal modulus. Multiple eigenvalues corresponding to linear divisors do not affect the speed of convergence or accuracy of eigenvalues in either the modified LR or double QR transformations. When the eigenvalues of equal modulus are not equal, these two algorithms also converge, but the eigenvalues seem to be somewhat less accurate for some matrices. When eigenvalues of equal modulus correspond to a matrix with non-linear divisors, the LR algorithm does not, in general, give convergence to upper triangular form. Nevertheless, when the double QR transformation is applied, the eigenvalues converge at approximately the same rate for small matrices and somewhat slower for large matrices.

Another important case is that of a real matrix with some real eigenvalues and some complex conjugate eigenvalues. The matrix  $A_k$  tends to an upper triangular matrix except for a single subdiagonal element associated with each complex conjugate pair of eigenvalues. Each of these subdiagonal elements is associated with a  $2 \times 2$  matrix centered on the diagonal whose eigenvalues converge to a complex conjugate pair. This condition poses no problem for the double QR algorithm in which the  $a_{n-1,n-2}$  element is checked before each iteration. Eventually this  $2 \times 2$  matrix appears as a lower right principal submatrix, and the complex conjugate eigenvalues are determined. The modified LR algorithm does not converge for complex conjugate eigenvalues. Parlett (7) states that he is working on a variation of Wilkinson's modified LR algorithm for use on real matrices with complex eigenvalues. For the orthodox LR and QR transformations such  $2 \times 2$  blocks will occur on the diagonal, but are difficult to detect by an automatic procedure.

#### IV. RESULTS AND CONCLUSIONS

The results of this study indicate that the QR algorithm is a more successful method than the LR algorithm for finding the eigenvalues of real unsymmetric matrices. Several observations have been made from the results of experimentation with numerous test matrices.

The convergence properties of the modified LR algorithm, when applied to real matrices, are not satisfactory except when the eigenvalues of the matrix are all real. Another restriction is that for convergence to take place any matrix with equal eigenvalues must have linear divisors. Also, a few matrices which are originally well-conditioned with respect to their eigenvalues were found to become progressively more ill-conditioned at each iteration. It does not seem to be uncommon for the size of the off diagonal elements to steadily increase resulting in a matrix  $A_k$  which is no longer similar to the original matrix  $A_1$ . When this occurred, sometimes the process either failed to converge after 25 iterations or failed to converge on all the correct eigenvalues. An example of a matrix, which converged on one eigenvalue correctly then failed to converge on the others, is cited in Table 6 of Appendix B. This growth of the off diagonal elements does not necessarily affect the convergence to the eigenvalues.

When comparing the use of Gaussian elimination and Householder's method for the preliminary reduction of the matrix to upper Hessenberg form, it was noticed that when Gaussian elimination was used with the modified LR algorithm 18.7% of the matrices tested required one or two more iterations than for Householder's method. For the double QR algorithm, one more iteration was necessary in 20% of the matrices tested for convergence of

the first eigenvalue when using Householder's reduction rather than Gaussian elimination. But in many of these cases, the total number of iterations required for convergence of all the eigenvalues were found to be the same. There may be some significance in the fact that fewer iterations were necessary when the same type of similarity transformation was used for both the preliminary reduction and the iteration.

The selection of a method for reducing the original matrix generally does not appear to make any significant difference in the accuracy of the eigenvalues found. However, for the matrices in Tables 5 and 9 or Appendix B, the use of Gaussian elimination with the modified LR algorithm caused a loss of accuracy in some of the computed eigenvalues. No such difficulty was encountered with the double QR algorithm.

The use of  $\epsilon = 10^{-6}$  as a criterion for convergence seemed to give the best results. When  $\epsilon = 10^{-8}$  was used with the modified LR algorithm, the results were often more accurate, but in some cases it disturbed the accuracy of the convergence of the double QR algorithm.

A comparison was made of the time required for convergence of the algorithms to all the eigenvalues of a matrix, and it was observed that the use of Householder's method with either algorithm required more time than Gaussian elimination with the same algorithm. Since the same number of iterations are usually required when using either method of reduction with an algorithm, this time difference is expected, because Householder's method requires  $5/3n^3$  multiplications as compared to  $5/6n^3$  for Gaussian elimination.

The total number of iterations in the convergence to the eigenvalues must be taken into consideration when comparing the times required by the modified LR and double QR algorithms. As the dimension of the test matrices used was increased, the number of iterations required for convergence by the modified LR increased at a faster rate than for the double QR. When the total number of iterations is the same for both algorithms, the double QR requires more time. This is again due to the number of multiplications involved in one iteration.

The modified LR transformation is much easier to apply, but because of its possible numerical instability and other restrictions, its value seems limited. Therefore, the double QR algorithm was found to be the more successful method for dealing with the real unsymmetric eigenvalue problem.

## APPENDIX A

\*FANDK1605

```

C      MODIFIED LR
C      USING SHIFTS (DETERMINED BY LOWER SUBMATRIX) AND DEFLATION
C      USE FOR REAL NONSYMMETRIC MATRICES WITH REAL EIGENVALUES
      DIMENSION A(20,20),SP(20),F(20,20),FN(20),KRW(20)
      5 READ 200, N
      READ 300,((A(I,J),J=1,N),I=1,N)
      SW1=0.0
      IT=0
      EPSLN=10.E-6
C      HOUSEHOLDER-REDUCTION OF NON-SYMMETRIC MATRICES TO UPPER-
C      HESSENBERG FORM
C      USING P=I-UUT/2K**2
      L=N-1
      DO 395 I=2,L
      SMS=0.0
      DO 350 K=I,N
350 SMS=SMS+A(K,I-1)**2
      S=SQRTF(SMS)
      IF(A(I,I-1)) 352,353,353
352 SGN=-1.
      GO TO 354
353 SGN=1.
354 X=A(I,I-1)+S*SGN
      A(I,I-1)=X
      BKSQ=SGN*S*X
      MP=I+1
      DO 355 K=I,N
355 SP(K)=A(K,I-1)/BKSQ
      SP(I-1)=-SGN*S
      MR=I-1
      DO 370 IK=I,N
      P=0.0
      DO 360 J=I,N
360 P=P+A(J,I-1)*A(J,IK)
      DO 365 JI=I,N
365 F(JI,IK)=A(JI,IK)-SP(JI)*P
      DO 366 J=1,MR
366 F(J,IK)=A(J,IK)
370 CONTINUE
      DO 390 KI=1,N
      Q=0.0
      DO 375 K=I,N
375 Q=Q+F(KI,K)*A(K,I-1)
      DO 380 K=I,N
380 A(KI,K)=F(KI,K)-Q*SP(K)
390 CONTINUE
      DO 391 J=MP,N
391 A(J,MR)=0.0
      A(I,MR)=SP(MR)
395 CONTINUE

```



```

DO 393 JK=1,N
393 PRINT 400, (A(JK,J),J=1,N)
C   CHOOSE ORIGIN SHIFTS
181 IF(ABSF(A(N,N-1))-EPSLN) 182,100,100
182 PRINT 400,A(N,N)
    N=N-1
    IT=0
    IF(N-2)190,185,181
185 SW1=1.0
    GO TO 100
190 PRINT 400,A(N,N)
    GO TO 195
100 B=-A(N,N)-A(N-1,N-1)
    C=A(N-1,N-1)*A(N,N)-A(N,N-1)*A(N-1,N)
    D=B**2-4.*C
    E=SQRTE(ABSF(D))/2.
    G=-B/2.
    IF(D)105,110,110
105 SH=G
    GO TO 125
110 P=G+E
    Q=G-E
    IF(SW1) 114,114,112
112 PRINT 400,P
    PRINT 400,Q
    GO TO 195
114 IF(ABSF(P-A(N,N))-ABSF(Q-A(N,N)))115,120,120
115 SH=P
    GO TO 125
120 SH=Q
125 DO 130 I=1,N
130 A(I,I)=A(I,I)-SH
C   CHECK SUBDIAGONAL ELEMENTS FOR DECOMPOSITION OF MATRIX
    NP=N-2
    KP=NP
    MN=N-1
    DO 132 MP=1,NP
    I=MN-MP
    IF (ABSF(A(I+1,I))-10.E-8) 134,134,131
131 IF (ABSF((A(I+1,I)*A(I+2,I+1))/A(I+1,I+1))-10.E-8) 134,134,132
132 KP=I
134 CONTINUE
C   ITERATE MODIFIED LR
    ML=N-1
    DO 152 KR=KP,ML
    KRW(KR)=KR
    MP=KR+1
    IF (A(KR+1,KR)-A(KR,KR))145,145,135
135 KRW(KR)=KR+1
    DO 140 J=KR,N
    SV=A(KR+1,J)
    A(KR+1,J)=A(KR,J)
140 A(KR,J)=SV
145 FN(KR)=A(KR+1,KR)/A(KR,KR)

```

```
      DO 150 J=KR,N
150  A(MP,J)=A(MP,J)-FN(KR)*A(KR,J)
152  CONTINUE
      DO 175 KR=KP,ML
      IF(KRW(KR)-KR) 165,165,155
155  DO 160 J=1,N
      SV=A(J,KR+1)
      A(J,KR+1)=A(J,KR)
160  A(J,KR)=SV
165  DO 170 J=1,N
170  A(J,KR)=A(J,KR)+FN(KR)*A(J,KR+1)
175  CONTINUE
      IT=IT+1
      PRINT 500,IT
      DO 180 I=1,N
      A(I,I)=A(I,I)+SH
180  PRINT 400, (A(I,J),J=1,N)
      IF(IT-25)181,181,194
194  PRINT 600
195  GO TO 5
196  CALL EXIT
200  FORMAT (15)
600  FORMAT (5X,22HPROCESS NOT CONVERGING)
300  FORMAT (7E10.2)
400  FORMAT (1X,6F18.8,/,4X,6F18.8)
500  FORMAT(/5X,I3,10HITERATIONS,/)
      END
```

## APPENDIX A (Cont'd)

\*FANDK1605

```

C      DOUBLE QR
      DIMENSION A(20,20),GAMA(3),RHO(3),FN(20)
      5 READ 800,N
      READ 300,((A(I,J),J=1,N),I=1,N)
      IZ=0
      EPSLN=10.E-6
C      GAUSSIAN ELIMINATION - UPPER HESSENBERG
      ML=N-2
      DO 345 KR=1,ML
      CK=ABSF(A(KR+1,KR))
      KOW=KR+1
      MP=KR+2
      DO 315 I=MP,N
      IF(ABSF(A(I,KR))-CK) 315,310,310
310 CK=ABSF(A(I,KR))
      KOW=I
315 CONTINUE
      DO 320 J=KR,N
      SV=A(KR+1,J)
      A(KR+1,J)=A(KOW,J)
320 A(KOW,J)=SV
      DO 325 I=MP,N
325 FN(I)=A(I,KR)/A(KR+1,KR)
      DO 330 J=KR,N
      US=A(KR+1,J)
      DO 330 I=MP,N
330 A(I,J)=A(I,J)-FN(I)*US
      DO 335 I=1,N
      SV=A(I,KR+1)
      A(I,KR+1)=A(I,KOW)
335 A(I,KOW)=SV
      DO 340 J=MP,N
      DO 340 I=1,N
340 A(I,KR+1)=A(I,KR+1)+FN(J)*A(I,J)
345 CONTINUE
      DO 350 I=1,N
350 PRINT 700, (A(I,J),J=1,N)
C      CHOOSE ORIGIN SHIFTS
      SW1=0.0
      SW2=0.0
      SWC=0.0
      10 IF(ABSF(A(N-1,N-2))-EPSLN) 25,15,15
      15 IF(ABSF(A(N,N-1))-EPSLN) 20,75,75
      20 PRINT 400, A(N,N)
      N=N-1
      GO TO 40
      25 IF(ABSF(A(N,N-1))-EPSLN) 35,30,30
      30 SW2=1.0
      GO TO 75
      35 PRINT 400,A(N,N)

```

```

        PRINT 400,A(N-1,N-1)
        N=N-2
40 IF(N-2) 50,50,45
45 GO TO 10
50 IF(N-1) 55,55,60
55 PRINT 400,A(N,N)
    GO TO 275
60 SW1=1.0
    IF(ABSF(A(N,N-1))-EPSLN) 70,75,75
70 PRINT 400,A(N,N)
    PRINT 400,A(N-1,N-1)
    GO TO 275
C    FIND EIGENVALUES OF LOWER SUBMATRIX
75 B=-A(N,N)-A(N-1,N-1)
    C=A(N-1,N-1)*A(N,N)-A(N,N-1)*A(N-1,N)
    D=B**2-4.*C
    E=SQRTF(ABSF(D))/2.
    G=-B/2.
    IF(D) 80,85,85
80 RTPD=G**2+E**2
    SWC=1.0
    GO TO 90
85 RTPD=G**2-E**2
90 RTSM=2.*G
    IF(SW1) 95,95,105
95 IF(SW2) 128,128,100
100 SW2=0.0
    N=N-2
105 IF(SWC) 110,110,120
110 RT1=G+E
    RT2=G-E
    PRINT 400, RT1
    PRINT 400, RT2
    IF(SW1) 40,40,115
115 SW1=0.0
    GO TO 275
120 SWC=0.0
    PRINT 500,G,E
    PRINT 600,G,E
    IF(SW1) 40,40,125
125 SW1=0.0
    GO TO 275
128 NP=N-3
126 MN=N-2
    KP=NP
    IF(N-3) 127,127,129
C    CHECK SUBDIAGONAL ELEMENTS FOR DECOMPOSITION
127 KP=1
    KQ=1
    GO TO 155
129 DO 140 MP=1,NP
    I=MN-MP
    IF(ABSF(A(I+1,I))-10.E-8) 145,145,130
130 IF(ABSF(A(I+1,I)*A(I+2,I+1)*(ABSF(A(I+1,I+1)+A(I+2,I+2))-FRTSM)+ABSF

```

```

      2(A(I+3,I+2)))/(A(I+1,I+1)*(A(I+1,I+1)-RTSM)+A(I+1,I+2)*A(I+2,I+1)+
      3RTPD))-10.E-8)145,145,140
140 KP=I
145 CONTINUE
      NM=KP+1
      KQ=KP
      DO 150 MP=1,KP
        I=NM-MP
        IF (ABSF(A(I+1,I))-10.E-8)155,155,150
150 KQ=I
155 CONTINUE
      NL=N-1
      PRINT 850,KP,KQ
C      ITERATE DOUBLE QR
      DO 265 I=KP,NL
        IF(I-KP)165,160,165
160 GAMA(1)=A(KP,KP)*(A(KP,KP)-RTSM)+A(KP,KP+1)*A(KP+1,KP)+RTPD
      GAMA(2)=A(KP+1,KP)*(A(KP,KP)+A(KP+1,KP+1)-RTSM)
      GAMA(3)=A(KP+1,KP)*A(KP+2,KP+1)
      A(KP+2,KP)=0.0
      GO TO 180
165 GAMA(1)=A(I,I-1)
      GAMA(2)=A(I+1,I-1)
      IF(I-(N-2)) 170,170,175
170 GAMA(3)=A(I+2,I-1)
      GO TO 180
175 GAMA(3)=0.0
180 FK=SQRTF(GAMA(1)**2+GAMA(2)**2+GAMA(3)**2)
      IF(GAMA(1))185,190,190
185 FK=-FK
190 IF(FK)200,195,200
195 RHO(1)=0.0
      RHO(2)=0.0
      ALPHA=2.
      GO TO 205
200 RHO(1)=GAMA(2)/(GAMA(1)+FK)
      RHO(2)=GAMA(3)/(GAMA(1)+FK)
      ALPHA=2./(1.+RHO(1)**2+RHO(2)**2)
205 IF(I-KQ)210,225,210
210 IF(I-KP)215,220,215
215 A(I,I-1)=-FK
      GO TO 225
220 A(I,I-1)=-A(I,I-1)
225 DO 240 J=I,N
      ZN=ALPHA*(A(I,J)+RHO(1)*A(I+1,J))
      LP=N
      IF(I-(N-2))230,230,235
230 ZN=ZN+ALPHA*(RHO(2)*A(I+2,J))
      A(I+2,J)=A(I+2,J)-RHO(2)*ZN
      LP=I+2
235 A(I,J)=A(I,J)-ZN
240 A(I+1,J)=A(I+1,J)-RHO(1)*ZN
243 DO 255 J=KQ,LP

```

```

      ZN=ALPHA*(A(J,I)+RHO(1)*A(J,I+1))
      IF(I-(N-2)) 245,245,250
245  ZN=ZN+ALPHA*(RHO(2)*A(J,I+2))
      A(J,I+2)=A(J,I+2)-RHO(2)*ZN
250  A(J,I)=A(J,I)-ZN
255  A(J,I+1)=A(J,I+1)-RHO(1)*ZN
      IF(I-(N-3)) 260,260,265
260  ZN+ALPHA*RHO(2)*A(I+3,I+2)
      A(I+3,I)=-ZN
      A(I+3,I+1)=-RHO(1)*ZN
      A(I+3,I+2)=A(I+3,I+2)-RHO(2)*ZN
265  CONTINUE
      DO 270 IJ=1,N
270  PRINT 700,(A(IJ,J),J=1,N)
      GO TO 10
275  CONTINUE
      IZ=IZ+1
      IF(IZ-10) 5,5,280
280  CALL EXIT
300  FORMAT (7E10.2)
400  FORMAT (5X,F18.8)
500  FORMAT (5X,F12.8,2H-I,F12.8)
600  FORMAT (5X,F12.8,2H+I,F12.8)
800  FORMAT (15)
750  FORMAT (5F18.8)
700  FORMAT (1X,6F18.8,/,4X,6F18.8)
850  FORMAT (5X,2HP=,F14.4,5X,2HQ=,F14.4,/)
      END

```

## APPENDIX B

## Numerical Results

Given below are some of the results obtained using the programs listed in Appendix A. The column headed O indicates the order in which the eigenvalues were found. The column headed I indicates the number of iterations required. The G and H indicate Gaussian elimination and Householder's method respectively were used for the preliminary reduction of the matrix. TIME does not include compile time.

TABLE I

$$\begin{bmatrix} .987 & .400 & -.487 \\ -.079 & .500 & .479 \\ .082 & .400 & .418 \end{bmatrix}$$

CORRECT	COMPUTED, $\epsilon = 10^{-6}$											
	LR-H			LR-G			QR-H			QR-G		
		O	I		O	I		O	I		O	I
.905	.90501276	3	0	.90504939	3	0	.90500001	3	0	.90500004	3	0
.900	.89998723	1	2	.89995063	1	2	.89999996	1	3	.89999984	1	3
.100	.09999999	2	0	.09999999	2	0	.10000001	2	0	.10000010	2	0
TIME	7"			5"			12"			12"		

TABLE II

$$\begin{bmatrix} -9 & -2 & -9 \\ -13 & -2 & -12 \\ 16 & 4 & 16 \end{bmatrix}$$

CORRECT	COMPUTED, $\epsilon = 10^{-6}$											
	LR-H	O	I	LR-G	O	I	QR-H	O	I	QR-G	O	I
1	1.00000169	1	4	1.00000145	1	4	.99999856	1	4	.99999944	1	4
2	2.00056091	3	0	2.00050652	3	0	2.00000071*2	0		2.00000027*2	0	
2	1.99943739	2	0	1.99949201	2	0	2.00000071*3	0		2.00000027*3	0	
TIME	10"			10"			14"			13"		

\* Imaginary parts  $\pm .000XXXXX$   
 Reduces to  $\pm .000000XX$  when  $\epsilon = 10^{-7}$

TABLE III

-5.509882	1.870086	.422908	.008814
.287865	-11.811654	5.711900	.058717
.049099	4.308033	-12.970687	.229326
.006235	.269851	1.397369	-17.596207

CORRECT	COMPUTED, $\epsilon = 10^{-6}$																							
	LR-H			0 I			LR-G			0 I			QR-H			0 I			QR-G			0 I		
-17.86303	-17.86326			1 3			-17.86326			1 3			-17.86326			2 0			-17.86326			2 0		
-17.15266	-17.15242			2 1			-17.15242			2 1			-17.15242			1 2			-17.15242			1 3		
-7.57405	-7.57404			4 0			-7.57404			4 0			-7.57404			4 0			-7.57404			4 0		
-5.29870	-5.29869			3 0			-5.29869			3 0			-5.29869			3 0			-5.29869			3 0		
TIME	15"						14"						15"						18"					

TABLE IV

1.022551	.116069	-.287028	-.429969
.228401	.742521	-.176368	-.283720
.326141	.097221	.197209	-.216487
.433864	.148965	-.193686	.006472

CORRECT	COMPUTED. $\epsilon = 10^{-6}$							
	LR-H		LR-G		QR-H		QR-G	
	O	I	O	I	O	I	O	I
.667483	.6674827	1 0	.6674827	1 0	.6674827	1 0	.6674827	1 0
.667483	.6674828	4 0	.6674828	4 0	.6674828	3 0	.6674828	3 0
.346148	.3461485	2 4	.3461483	2 4	.3461485	2 3	.3461483	2 3
.287639	.2876388	3 0	.2876390	3 0	.2876388	4 0	.2876390	4 0
TIME	15"		10"		13"		12"	



TABLE V

.81321	-.00013	.00014	.00011	.00021
-.00013	.93125	.23567	.41235	.41632
.00014	.23567	.18765	.50632	.30697
.00011	.41235	.50632	.27605	.46322
.00021	.41632	.30697	.46322	.41931

CORRECT	COMPUTED, $\epsilon = 10^{-6}$							
	LR-H	O I	LR-G	O I	QR-H	O I	QR-G	O I
1.67828	1.678279	5 0	1.678046	5 0	1.678279	5 0	1.678281	5 0
.81321	.813210	4 0	.813210	4 0	.813210	4 0	.813210	4 0
.41985	.419850	3 1	.422321	3 1	.419852	3 1	.419850	3 1
.01521	.015210	2 2	.015210	1 5	.015210	2 0	.015210	2 0
-.29908	-.299082	1 3	-.301318	2 0	-.299082	1 3	-.299082	1 3
TIME	23"		22"		29"		28"	

TABLE VI

-.05811	-.11695	.51004	.31330
-.04291	.56850	-.07041	-.68747
-.01652	.38953	.01203	-.52927
.06140	.32179	-.22094	-.42448

CORRECT	COMPUTED, $\epsilon = 10^{-6}$							
	LR-H	O I	LR-G	O I	QR-H	O I	QR-G	O I
.13512787	.13512787	1 5	.13512787	1 6	.13512744	2 0	.13512228	1 3
-.25660142	-		-		-.25660141	4 0	-.25661046	4 0
.32185293	-		-		.32185293	3 0	.32185883	2 2
-.10243938	-		-		-.10243896	1 4	-.10243065	3 0
TIME	20"		17"		23"		23"	

TABLE VII

26	-54	4
13	-28	3
26	-56	5

CORRECT	COMPUTED, $\epsilon = 10^{-6}$			
	QR-H		QR-G	
1+5i	1.00000036 + 4.99999976i	3 0	1.00000054 + 4.99999999i	3 0
1-5i	1.00000036 - 4.99999976i	2 0	1.00000054 - 4.99999999i	2 0
1	.99999927	1 4	.99999891	1 4

TIME

15"

15"

TABLE VIII

253	121	66	11	11	0
121	96	-19	71	-24	7
66	-19	137	-117	73	-14
11	71	-117	152	-82	21
11	-24	73	-82	57	-14
0	7	-14	21	-14	7

CORRECT	COMPUTED, $\epsilon = 10^{-6}$							
	LR-H		LR-G		QR-H		QR-G	
332.849					332.849319 4 1		332.849319 6 0	
332.849			332.849319 3 0		332.849319 3 1		332.849316 3 0	
15.618					15.617767 6 0		15.617769 5 0	
15.618	15.617767 2 1		15.617767 2 2		15.617767 2 0		15.617767 2 0	
2.533					2.532912 5 0		2.532914 4 1	
2.533	2.532912 1 2		2.532912 1 2		2.532912 1 2		2.532912 1 2	

TIME

3'5"

42"

33"

30"

TABLE IX

12	11	10	9	8	7	6	5	4	3	2	1
11	11	10	9	8	7	6	5	4	3	2	1
10	10	10	9	8	7	6	5	4	3	2	1
9	9	9	9	8	7	6	5	4	3	2	1
8	8	8	8	8	7	6	5	4	3	2	1
7	7	7	7	7	7	6	5	4	3	2	1
6	6	6	6	6	6	6	5	4	3	2	1
5	5	5	5	5	5	5	5	4	3	2	1
4	4	4	4	4	4	4	4	4	3	2	1
3	3	3	3	3	3	3	3	3	3	2	1
2	2	2	2	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1	1	1

CORRECT 8 digits	COMPUTED, $\epsilon = 10^{-6}$											
	LR-H			LR-G			QR-H			QR-G		
		O	I		O	I		O	I		O	I
63.409139	63.40913894	12	0	63.40913891	12	0	63.40913894	12	0	63.40913894	12	0
7.1201222	7.12012218	11	0	7.12010641	11	0	7.12012217	11	0	7.12012219	11	0
2.6180340	2.16803393	10	1	2.61804981	10	0	2.61803398	10	0	2.61803401	10	0
1.3790212	1.37902123	9	1	1.37919096	9	1	1.37902118	9	0	1.37902133	9	1
.87074533	.87074535	8	1	.87057574	8	0	.87074532	8	1	.87074711	8	0
.61529474	.61529502	7	1	.61498698	7	1	.61529473	7	0	.61529561	7	1
.47045960	.47046013	6	1	.47603698	6	1	.47045959	6	1	.47045756	6	0
.38196601	.38196489	5	1	.37669448	5	0	.38196601	5	0	.38196536	5	1
.32555754	.32555464	4	2	.32561060	4	2	.32555754	4	1	.32555736	2	0
.28918975	.28941734	3	1	.29114627	3	2	.28918974	2	1	.28918902	4	0
.26648096	.26648087	1	5	.26648094	1	5	.26658095	1	4	.26658096	1	3
.25398978	.25376543	2	0	.25198187	2	0	.25398977	3	0	.25399049	3	1
TIME	3'13"			3'			4'8"			3'		

## BIBLIOGRAPHY

1. Faddeev, D. K. and Faddeeva, V. N., (1963) Computational Methods of Linear Algebra, Translated by Robert C. Williams, W. H. Freeman and Company, San Francisco and London p. 25-32, 475-478.
2. Francis, J. G. F., (1961) The QR Transformation - A Unitary Analogue to the LR Transformation - Part 1, The Computer Journal 4, p. 265-271.
3. Francis, J. G. F., (1961) The QR Transformation - Part 2, The Computer Journal 4, p. 332-345.
4. Henrici, Peter, (1958) The Quotient-Difference Algorithm, National Bureau of Standards Applied Mathematics Series 49, p. 23-46.
5. Henrici, P. and Watkins, Bruce O., (1965) Finding Zeros of a Polynomial by the Q-D Algorithm, Communications of the Association for Computing Machinery 8, p. 570-574.
6. Householder, A. S., (1961) On Deflating Matrices, Journal of the Society of Industrial and Applied Mathematics 9, p. 89-93.
7. Parlett, Beresford, (1964) The Development and Use of Methods of LR Type, Siam Review 6, p. 275-295.
8. Ralston, Anthony, (1965) A First Course in Numerical Analysis, McGraw-Hill, Inc., New York, New York p. 496-520.
9. Rutishauser, Heinz, (1958) Solution of Eigenvalue Problems With the LR Transformation, National Bureau of Standards Applied Mathematics Series 49, p. 47-81.
10. Wilkinson, J. H., (1959) Stability of the Reduction of a Matrix to Almost Triangular and Triangular Forms by Elementary Similarity Transformations, Journal of the Association for Computing Machinery 6, p. 336-359.
11. Wilkinson, J. H., (1960) Householder's Method for the Solution of the Algebraic Eigenproblem, The Computer Journal 3, p. 23-27.
12. Wilkinson, J. H., (1965) Convergence of the LR, QR, and Related Algorithms, The Computer Journal 8, p. 77-84.
13. Wilkinson, J. H., (1965) The Algebraic Eigenvalue Problem, Oxford University Press, London 264 p.

## VITA

The author was born on September 9, 1942, in Sault Ste. Marie, Michigan. She received her primary education in Manistique, Michigan, and attended high school there and in Rolla, Missouri. She attended college at Park College in Parkville, Missouri, and the University of Missouri at Rolla in Rolla, Missouri, receiving a Bachelor of Arts degree from Park College in December 1963. Since September 1964, she has been enrolled in the University of Missouri at Rolla as a graduate student in Computer Science.